```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> #Load the necessary libraries
> library(boot)

Attaching package: ‘boot’

The following object is masked _by_ ‘.GlobalEnv’:

    motor

> #Load the data from CSV file
> gpa <- read.csv("C://Users//yxa210024//Desktop//Masters//spring2023//Stats for DS//mini_project
4//gpa.csv")
> #Attach the data for easy access to variables
> attach(gpa)
The following object is masked _by_ .GlobalEnv:

    gpa

> #Generate scatter plots
> plot(gpa ~ act, data = gpa, pch = 20, main = "Scatterplot")
> #Calculate the Pearson correlation coefficient between gpa and act
> Cor_relation <- cor(gpa, act)
> #Define a function for bootstrap resampling
> Correlation_Resampling_function <- function(corr, indices){
+ cor(corr[indices, 1], corr[indices, 2])
+ }
> #Define a function to get the bootstrapped samples
> Corr_boots_samp <- function(corr, i = c(1:n))
+ {
+ c <- corr[i,]
+ return(c)
+ }
> #Set the number of bootstrap replicates
> Boot_replicates <- 1000
> set.seed(1000)
> #Perform bootstrap resampling for correlation between gpa and act
> Cor_boot_resample <- boot(gpa, Correlation_Resampling_function, R = Boot_replicates)
> #Display the results of bootstrap resampling
> Cor_boot_resample

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = gpa, statistic = Correlation_Resampling_function,
    R = Boot_replicates)
```

```
Bootstrap Statistics :
     original      bias    std. error
t1* 0.2694818 0.008486976   0.1092285
> #Calculate the 95% percentile confidence interval for correlation between gpa and act
> Boot_cofidence_interval(Cor_boot_resample, conf = 0.95, type = "perc")
Error in Boot_cofidence_interval(Cor_boot_resample, conf = 0.95, type = "perc") :
  could not find function "Boot_cofidence_interval"
> boot.ci(Cor_boot_resample, conf = 0.95, type = "perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = Cor_boot_resample, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.0575,  0.4966 )
Calculations and Intervals on Original Scale
>
>
> > Boot_replicates <- 10000
Error: unexpected '>' in ">"
> > set.seed(1000)
Error: unexpected '>' in ">"
> > #Perform bootstrap resampling for correlation between gpa and act
Error: unexpected '>' in ">"
> > Cor_boot_resample <- boot(gpa, Correlation_Resampling_function, R = Boot_replicates)
Error: unexpected '>' in ">"
> > #Display the results of bootstrap resampling
Error: unexpected '>' in ">"
> > Cor_boot_resample
Error: unexpected '>' in ">"
> Boot_replicates <- 10000
> set.seed(1000)
> Cor_boot_resample <- boot(gpa, Correlation_Resampling_function, R = Boot_replicates)
> Cor_boot_resample

ORDINARY NONPARAMETRIC BOOTSTRAP


Call:
boot(data = gpa, statistic = Correlation_Resampling_function,
    R = Boot_replicates)


Bootstrap Statistics :
     original      bias    std. error
t1* 0.2694818 0.004057848   0.1049293
> boot.ci(Cor_boot_resample, conf = 0.95, type = "perc")
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates

CALL :
boot.ci(boot.out = Cor_boot_resample, conf = 0.95, type = "perc")

Intervals :
Level     Percentile
95%   ( 0.0701,  0.4791 )
Calculations and Intervals on Original Scale
>
```